

$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ -1 & i \end{pmatrix}$

Stylesheet Version 1.0

## Cross Reference to Related Applications

## Background of Invention

[0002] A problem with current KVM switches is that if a USB peripheral, such as a printer, is connected to the switch, data flow is interrupted to that peripheral when the switch is changed.

[0004] The most current information concerning the USB standard is maintained at the following site on the World Wide Web: <http://www.usb.org>. Available at that source

are the Universal Serial Bus (USB) Specification, the USB HID Usage Supplement, and the Device Class Definition for Human Interface Devices (HID).

[0005] The HID class consists primarily of devices that are used by humans to control the operation of computer systems. Typical examples of HID class devices include: keyboards and pointing devices—for example, standard mouse devices, trackballs, and joysticks; front-panel controls—for example: knobs, switches, buttons, and sliders; controls that might be found on devices such as telephones, VCR remote controls, games or simulation devices—for example: data gloves, throttles, steering wheels, and rudder pedals; and devices that may not require human interaction but provide data in a similar format to HID class devices—for example, bar-code readers, thermometers, or voltmeters.

[0006] What is needed is a KVM switch that is also a peripheral sharing switch, which would allow all the computers connected to the switch to share any USB peripheral devices without interruption of data flow to that peripheral when the switch is changed, and which would switch the KVM channels and peripheral channels to a common computer or to different computers either asynchronously or synchronously.

## Summary of Invention

[0007] The present invention meets this need by providing a KVM switch that is also a peripheral sharing switch, which allows all the computers connected to the switch to share any USB peripheral devices, and which can switch the KVM channels and peripheral channels to a common computer or to different computers either asynchronously or synchronously without interruption of data flow to that peripheral when the switch is changed. USB peripherals may be synchronously or asynchronously switched with the KVM channels.

[0008] The present invention, USB KVMP, allows multiple USB keyboards and multiple USB mice communicating with a common computer at the same time. The USB KVMP provides multiple USB peripheral channels to users and those channels can be connected with all kinds of USB peripherals, such as a USB hub, USB printer, USB scanner, USB camera, etc.

[0009] A signal switch is disclosed for sharing a video monitor, a plurality of console

devices and one or more than one peripheral device in any of a plurality of computer systems, comprising a CPU comprising a first memory for storing a management program for managing the signal switch; a hub switch module connected to the CPU and configured to communicate with any of the plurality of computer systems, and the one or more than one peripheral device, such that a signal passing from the hub switch module to the one or more than one peripheral device emulates origination from a computer; a device control module for emulating the video monitor and plurality of console devices, connected to the CPU and the hub switch module; a host control module connected to the CPU and configured to communicate with the plurality of console devices; and a video control module connected to the CPU and configured to communicate with a video monitor device.

[0010] In a further embodiment, the signal switch further comprises an OSD control device connected to the CPU and the video control module.

[0011] In a preferred embodiment the host control module comprises a root hub, the plurality of console devices and one or more than one peripheral device are USB compliant, and the console devices comply with an industry standard specification.

[0012] A method is described for sharing a video monitor, a plurality of console devices compliant with an industry standard and one or more than one peripheral device in any of a plurality of computer systems through a signal switch, comprising: initializing the signal switch; emulating one or more of the console devices according to the industry standard; enumerating ports of a root hub; determining whether any downstream ports exists, and if so, enumerating the downstream ports; determining whether any of the plurality of console devices is connected to the root hub, or any downstream ports, and if so, then enumerating each connected device; determining whether any of the connected devices is compliant with the industry standard; enumerating each complaint connected device and parsing any data from such device; and repeatedly polling to determine whether any of the plurality of console devices, any of the one or more than one peripheral device, or any downstream port, has been plugged or unplugged, and if so, resetting control.

[0013] In one preferred embodiment, the management program comprises steps for managing the signal switch, and the described method.

[0014] A signal switch is described for sharing one or more console devices and one or more peripheral devices in any of a plurality of computer systems, comprising: a first channel for connecting a selected console device from the one or more console devices to a first selected computer system from the plurality of computer systems; a second channel connecting the first selected computer system to a selected peripheral device from the one or more peripheral devices, the second channel having a data flow between the first selected computer system and the selected peripheral device; a third channel for connecting the selected console device to a second selected computer system from the plurality of computer systems; and means for switching the selected console device between the first channel and the third channel without interruption of the data flow through the second channel between the first selected computer system and the selected peripheral device.

## Brief Description of Drawings

[0015] These and other features, aspects, and advantages of the present invention will become better understood with reference to the following description, appended claims, and accompanying drawings, where:

[0016] Figure 1 is a system diagram of one possible use of the use of signal switch of Figure 4.

[0017] Figure 2 is a system diagram of another possible use of the use of signal switch of Figure 4.

[0018] Figure 3 is a system diagram of another possible use of the use of signal switch of Figure 4.

[0019] Figure 4 is a block diagram of the signal switch of the present invention.

[0020] Figure 5 is a function block diagram of process useable as a management control program for the signal switch of Figure 4.

## Detailed Description

[0021]

The present invention is directed to a signal switch for sharing one or more than one video monitor, keyboard, mouse and peripheral device. The signal switch of the

present invention is the combination of a KVM (keyboard video mouse) switch and a peripheral sharing switch. KVM switches are well known in the art with reference to this disclosure. The signal switch according to the present invention allows all computers connected to the switch to share all USB peripheral devices connected to the switch.

[0022] The signal switch can either asynchronously or synchronously switch KVM channels and peripheral channels to a common computer or different computer. In other words, the KVM channels and peripheral channels may be switched together (synchronously) or separately (asynchronously).

[0023] The present invention utilizes USB emulation programs to emulate the HID (Human Interface Devices) specification, thereby enabling a switch to communicate with the USB ports of PCs, and through USB host emulation programs, enable a switch to communicate with USB devices such as USB keyboards and USB mice, and a USB hub. In other words, the use of emulation makes a switch appear as a computer to peripheral devices, and enables the switch to communicate with USB devices or USB PCs at the same time.

[0024] Although an embodiment is described with reference to the current HID specification, it will be apparent to those skilled in the art with reference to this disclosure that the invention may be implemented with any specification for device interfacing.

[0025] Figure 1 illustrates one of the many ways in which the present invention may be used. In a simple configuration, a signal switch 10 according to the present invention permits a user to control a plurality of computer systems 12 (represented by first computer 121 and second computer 122), and USB compatible peripherals 20 (as shown by first printer 22) with a monitor 14, first keyboard 16, and first mouse 18. In this example, control of the peripheral first printer 22 can be maintained by computer 121 even while monitor 14, first keyboard 16 and first mouse 18 are controlling computer 122. This is because switch 10 can switch between a first channel (not shown) connecting monitor 14, first keyboard 16 and first mouse 18 to first computer 121 and a third channel (not shown) connecting monitor 14, first keyboard 16 and first mouse 18 to second computer 122, while maintaining a second channel (not

shown) connecting first computer 121 to first printer 22 such that a first data flow between first computer 121 and first printer 22 is not interrupted.

[0026] With reference to Figure 2, a more robust use is shown permitting a user to control a larger number of computer systems 12 (represented by first computer 121, second computer 122, third computer 123, and fourth computer 124), and USB compatible peripherals 20 with monitor 14, first keyboard 16, and first mouse 18. Note that with the present invention it is possible to have one or more than one USB hub 24 also connected to signal switch 10, which in turn can connect to one or more than one further peripherals such as scanner 241, or one or more than one downstream hub 242, which in turn may be connected to one or more than one peripherals such as second printer 2421. It thus can be seen that any desired tree of hubs/peripherals can be connected to signal switch 10 and thus controlled synchronously or asynchronously by the one or more than one computer systems 12 under the management of a single set of monitor 14, first keyboard 16, and first mouse 18.

[0027] Thus, for example, the first data flow between first computer 121 and first printer 22, a second data flow between third computer 123 and scanner 241, and a third data flow between fourth computer 124 and second printer 2421 all could be maintained without interruption while keyboard 16 and mouse 18, and optionally monitor 14, are switched among computer systems 12.

[0028] Peripheral devices that can be connected with the present invention include but are not limited to a USB hub, printer, scanner and camera. Other examples of peripherals will be understood by one skilled in the art with reference to this disclosure.

[0029] With reference to Figure 3, in a still further use of the present invention, by way of example it is possible to connect multiple KM devices such as second keyboard 161, third keyboard 162, second mouse 181, second mouse 182, first composite KM device 191 and second composite KM device 192. Preferably, since a signal switch usually only has two USB console ports, one or more than one downstream hub may be connected, such as first downstream hub 171 and second downstream hub 172, to which may be connected one or more than one KM devices.

[0030] With reference to Figure 4, the signal switch 10 comprises a central processing unit ("CPU") 30 for managing the signal switch 10. A USB hub switch module 32 is connected to the CPU 30 and is configured to communicate with a plurality of computer systems 12 through first output ports 34. The USB hub switch module 32 is also configured to communicate with peripheral devices 20 through output ports 36. In Figure 4, four connections are shown between first output ports 34 and the USB hub switch module 32, but this is by way of example only. The number of connections will equal the number of computer systems 12 for which the signal switch 10 is configured.

[0031] Likewise, in Figure 4, two connections are shown between second output ports 36 and the USB hub switch module 32, but this is also by way of example only. The number of connections will equal the number of peripheral output ports for which the signal switch 10 is configured.

[0032] The USB hub switch module 32 is a bridge between peripheral devices 20 and computer systems 12 and allows the signal switch 10 to connect each of a plurality of computer systems to one or more than one peripheral device. Construction of a circuit suitable as a USB hub switch module is well known in the art with reference to this disclosure. In one preferred embodiment, the USB hub switch module includes 4 USB hubs and matrix analog switches which are controlled by CPU firmware. Texas Instruments(r) manufactures USB Hub chips that are suitable for this module, and the module can be constructed using Application Specific Integrated Circuit (ASIC) design methodology.

[0033] A USB device control module 38 for controlling signals is connected to the CPU 30 and the USB HUB switch module 32. The USB device control module 38 comprises USB device chips that are used to emulate the console devices, such as first keyboard 16 and first mouse 18, for the first output ports 34. In other words, by having a USB device chip emulate console devices attached to a first computer system, actual console devices may be switched to a second or different computer system, leaving any channels between the first computer system and peripherals connected, any data flow in such channels uninterrupted, and the first computer system still processing as if the actual console devices, now emulated, were still connected. These chips are

controlled by CPU 30 firmware. One device chip is required for each computer system 12. Construction of a circuit suitable as a USB device control module is well known in the art with reference to this disclosure.

[0034] In Figure 4, four connections are shown between the USB device control module 38 and the USB hub switch module 32, but this is by way of example only. The number of connections will equal the number of computer systems 12 for which the signal switch 10 is configured. In one preferred embodiment, the USB device control module includes 4 USB device chips and the chips are also controlled by CPU firmware. Philips Semiconductors(r) makes chips suitable for use in this module, and the module may also be constructed using Application Specific Integrated Circuit (ASIC) design methodology.

[0035] Referring to Figure 4, the CPU 30 comprises a first memory 40 for storing a management program 42 for managing the operation of the signal switch 10. A flowchart of the function of a portion of management program 42 suitable for use in the present invention is illustrated in Figure 5, and described below.

[0036] A USB host control module 44 is configured to communicate with a plurality of console devices and is connected to the CPU 30. Construction of a circuit suitable as a USB host control module is well known in the art with reference to this disclosure. The USB host control module 44 itself comprises a root hub 46 for communicating with one or more than one console devices or downstream hubs, through third output ports 48. The root hub is a USB compatible hub, which is well known in the art with reference to this disclosure.

[0037] As shown in Figure 3, one or more than one downstream hub may be connected to the signal switch 10, such as first downstream hub 171 and second downstream hub 172, communicating with root hub 46 through one of the third output ports 48. It is possible to daisy chain up to 127 devices by adding more downstream hubs.

[0038] Typically, third output ports 48 comprise two physical ports. Optionally, in a different embodiment, an internal additional console USB hub (not shown) may be connected to the root hub 46 for increasing the number of third output ports 48 available for connecting console devices. The CPU 30 firmware has the ability to



handle hub functions. Construction of firmware suitable to run the CPU according to the method disclosed herein is known in the art with reference to this disclosure.

[0039] An optional video/panel control device module 50 is connected to the CPU 30 for communicating with a variety of optional panel display devices 54 (shown in broken lines), such as light panels, LEDs, buttons, video controller switches, or other optional panel devices. If an on screen display ("OSD") function is provided, then an optional OSD control device module 56 (shown in broken lines) is connected to the CPU 30 and the video/panel control device module 50 for controlling and adjusting signals to an external monitor 14 through fourth output port 52. Construction of circuits suitable for use as a video/panel control device module or optional OSD control device module is known in the art with reference to this disclosure.

[0040] With reference to Figure 5, the process of a portion of a management program 42 suitable for use in the present invention may proceed as follows. In step 210 the signal switch 10 is initialized. In step 212, a root hub 46 is enumerated and any root hub message from the root hub 46 is saved. In step 214 it is determined whether any device is connected to the root hub 46. If a device is determined to be connected to the root hub 46, program process passes to step 216, else the process passes to step 226.

[0041] In step 216, each device determined to be connected to the root hub 46 is enumerated, and a device message from the USB device determined to be connected to the root hub 46 is parsed and saved in a suitable memory.

[0042] Next, in step 218, it is determined whether any downstream hub is connected to the root hub 46. If no downstream hub is determined to be connected to the root hub 46, the process proceeds to step 228. If a downstream hub is detected, the process passes to step 220. In step 220 each downstream hub determined to be connected to the root hub 46 is enumerated and a downstream port status of the hub is received. The process then proceeds to step 222.

[0043] In step 222, it is determined whether any device is existing in a port of the downstream hub. If no device is determined to be existing in a port of the downstream hub, the process proceeds to step 228. If, in step 222, a device is

determined to be existing in a port of a downstream hub, the process proceeds to step 224 where each such device is enumerated and a connecting device message is parsed and saved in a suitable memory. The process then proceeds to step 228.

[0044] In step 228, it is determined whether the connected device is a HID keyboard or mouse or other HID compliant device. As will be evident to those skilled in the art with reference to this disclosure, this step could test for compliance with any desired industry standard or device class. If the connected device is not such a device, the process proceeds to step 232 which is an error process which results in resetting a HOST control RAM in step 240. If the connected device is determined to be compliant, then in the next step 230, each such connected device is enumerated and its report data is parsed. The reason for the compliance test is that the switch has been constructed to emulate certain console devices which comply with an industry standard device class.

[0045] Control then passes to step 226 where the HUB switch module 32 is polled. In optional step 234 any panel or LED circuit is polled. In step 238 it is determined whether any downstream port is unplugged or plugged. If the downstream port is unplugged or plugged the process proceeds to step 240 where the HOST control RAM is reset. After resetting the HOST control RAM in step 240 the process returns to step 212. Else, the process proceeds to step 242 where the KVM devices are polled and the process returns to step 234.

[0046] The firmware controls the USB console devices and PC console device emulations. It is not necessary to control the USB peripherals, since the invention generates a path or channel between the USB peripherals and the linked PCs which is undisturbed by switching the channels between the complaint KM devices and the PCs.

[0047] Those skilled in the art with reference to this disclosure will be able to construct a USB emulation program suitable for use in the present invention by following the human interface device (HID) specification. The HID specification defines a quasi-language for the HID-compliant USB device to tell the host how to interpret the data that the host receives. This quasi-language is flexible and permits many ways to describe the same piece of functionality. The invention will be useable with other industry standard specifications that define a procedure to tell a host how to interpret

the data that the host receives from a standards-compliant device.

- [0048] Currently, a Device Class Definition for Human Interface Devices (HID) Firmware Specification-6/27/01, Version 1.11, is available at [http://www.usb.org/developers/data/devclass/HID1\\_11.pdf](http://www.usb.org/developers/data/devclass/HID1_11.pdf) and is incorporated herein. The most current version of the specification and other HID related information is maintained at <http://www.usb.org/developers/hidpage.html>.
- [0049] Using a bus analyzer, such as one available currently from Computer Access Technology Corp. it is possible to read the USB bus and analyze the data packets coming from a keyboard and a mouse. The HID specification provides interpretation of the packet meaning. By providing that all the console devices be complaint with the HID specification, or some future standard specification, it is only necessary to emulate such HID or standards-compliant devices.
- [0050] In a preferred embodiment the whole system programs, which include USB Hub switch control, USB device chip control (device emulation) and console device emulation (Host emulation) are all built in a CPU chip (Flash ROM or PROM). One skilled art with reference to this disclosure and following the referenced specifications will be able to write a USB emulation program suitable to make a switch appear as a PC to peripheral devices, and enable the switch to communicate with USB devices or USB PCs at the same time.
- [0051] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.